Lecture 5: Sources of Error

Nigel Shen cs5897@nyu.edu

February 2022

1 Written Homework 1 Hint

The problem we want to solve is the convergence of $f(y) = \frac{1}{2}(\frac{x}{y} + y)$.
Our goal is to:

1 Fix $y_0 > 0$

2 If $y_0 < \sqrt{x}$, check the region where y_1 falls in, and choose $[a, b]$ wisely so that the Contract Mapping Theorem can be applied.

2 Four Major sources of error

1. Truncation error
2. Termination error
3. Statistical error
4. Round off error (Not covered in this lecture)

3 Truncation Error

Let's compute a finite difference approximation to the derivative:

$$f(x+h) = f(x) + f'(x)h + \frac{f''(x)h^2}{2} + O(h^3)$$

Subtract $f(x)$ from both sides, divide by h, and rearrange, we get:

$$f'(x) = \frac{f(x+h) - f(x)}{h} + \frac{f''(x)h}{2} + O(h^2)$$

If we truncate and approximate $f'(x)$ with the $k = 1$ Taylor polynomial, we get:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

with truncation error:

$$\frac{f''(x)h}{2} + O(h^2)$$

This particular finite difference approximation is called a *forward difference*.

Exercise: With $x = \frac{\pi}{2}$, approximate $\frac{d}{dx} sin(x)$ at $x+h$ where $h = 0.1, 0.01, 0.001, ...$ What trend do you observe? Can you relate it to the values of $\frac{f''(x)}{2}$?

Def: The error in an approximation like this can be written as $Ch^p + O(h^p + 1)$, then the approximation is p^{th} order accurate.

Thus, we find that $f(x) \approx \frac{f(x+h) - f(x)}{h}$ is first-order accurate. If we did a *least square fit* of the errors, we would expect them to match Ch^1.

Now, in order to increase the accuracy, let's take the Taylor expansion with one more term:

$$f(x+h) = f(x) + f'(x)h + f''(x)\frac{h^2}{2} + f^{(3)}(x)\frac{h^3}{6} + O(h^4)$$

$$f(x-h) = f(x) - f'(x)h + f''(x)\frac{h^2}{2} - f^{(3)}(x)\frac{h^3}{6} + O(h^4)$$

Subtract them, we get:

$$f(x+h) - f(x-h) = 2f'(x)h + \frac{f^{(3)}(x)}{3}h^3 + O(h^4)$$

Rearrange terms, we get:

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{f^{(3)}(x)}{6}h^2 + O(h^3)$$

This is called a *central difference*. It is second-order accurate.

Exercise:
1. What happens if $f^{(3)}(x) = 0$?
2. $p(x) = ax^2 + bx + c$. Show that the central difference scheme is "exact": That there is no error.

4 Termination Error

We saw two different kinds of iterative schemes in previous lectures:

* $x_{n+1} = f(x_n)$, which solves $f(x) = x$ to find the fixed points;
* $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$ which solves $f(x) = 0$ (Newton's equation solving root-finding)

Exercise: The Secant method makes a linear approximation from x_n,x_{n-1},$f(x_n)$,$f(x_{n-1})$ at each step and finds that line's intersection with the x-axis to compute x_{n+1}. Newton's method uses f_n and f'_n. Write down a method similar to the secant method which makes a quadratic approximation using x_n, x_{n-1}, x_{n-2} and $f_n, f_{n-1}f_{n-2}$ and finds its intersection with the x-axis at each step.

Let's denote $e_n = x_n - x$ where x is the solution. This is the absolute error at each step. The method converges with order q accuracy if

$$\lim_{n\to\infty} \frac{e_{n+1}}{e_n^q} = \mu \in (0,1)$$

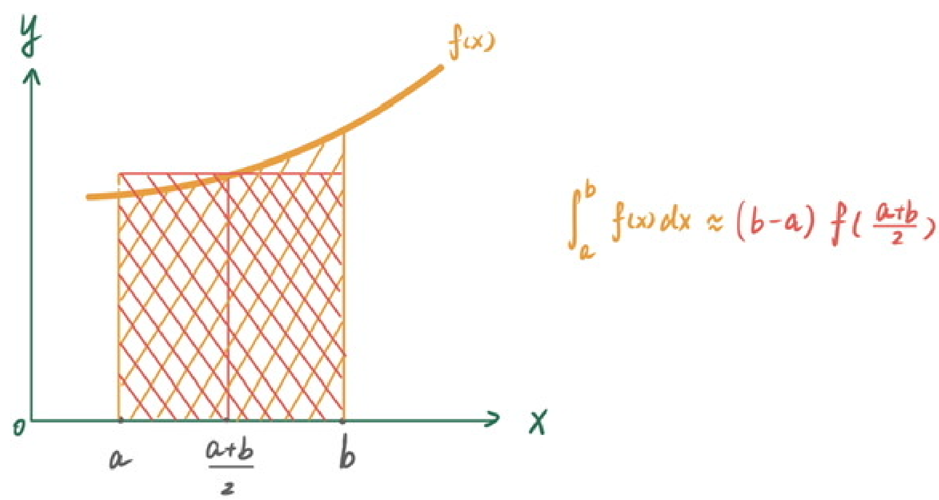
Note: e_n is as hard to compute as x_n. We don't know x. If we know e_n and x_n well, then we can find $x = x_n + e_n$ and we are done. We can instead look at $x_{n+1} - x_n$, which tells us if x_n converges.

5 Statistical Error

We'll see later in the class methods for approximating integrals ("numerical quadrature"). The idea comes from the derivation of Riemann integral:

$$\int_a^b f(x)dx \approx \sum_{i=1}^n w_i f(x_i)$$

Mid-point Estimation: If the interval $[a, b]$ is not big, we might want to use Mid-point estimation: $\int_a^b f(x)dx \approx f(\frac{a+b}{2})(b-a)$



Exercise: Derive an error estimate for the midpoint rule applied to $\int_a^b f(x)dx$

assuming $b - a = O(h), h > 0$ and using a Taylor expansion.

For integrating multidimensional functions we can use tensor product quadrature:

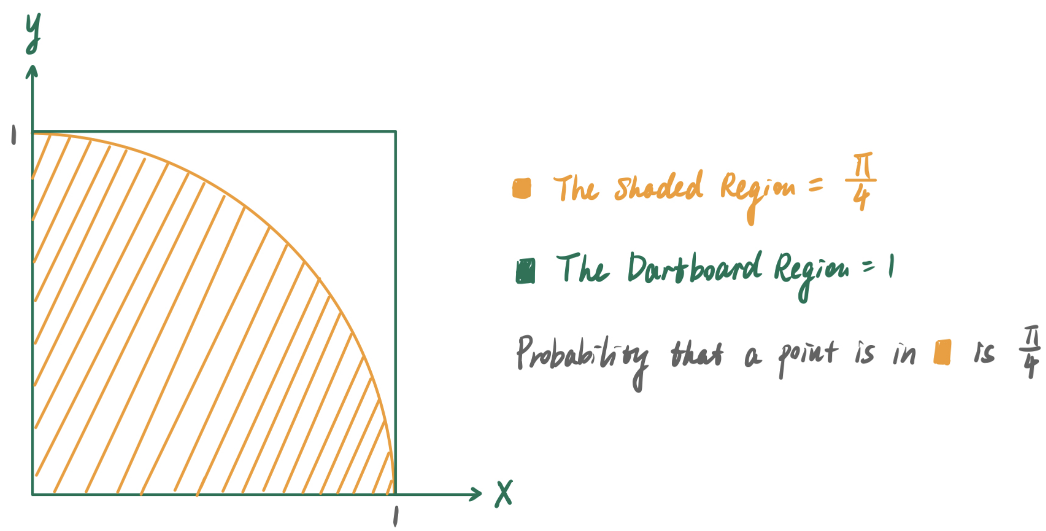
$$\int_{a_1}^{b_1} \int_{a_0}^{b_0} f(x, y) dx dy \approx \sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} f(x_i, y_j)$$

Let's say we want to integrate a d-dimensional function using a $1D$ n-points quadrature rule, then the time complexity is $O(n^d)$. Therefore, instead people use *Monte-Carlo* methods.

Monte-Carlo Method: The idea is to generate a sequence of random approximations x_n approximating x. The error will frequency decay like $O(\frac{1}{\sqrt{n}})$ independent of dimension.

Example: Approximating π

The idea is to make a dartboard:



Let (x_n, y_n) be the position of the n^{th} dart throw, where $x_n, y_n \sim Uniform([0, 1])$ and x_n, y_n are independent variables, so that our throws are independently and identically distributed.
After each throw we check whether it is in the shaded region:

$$h_n = \begin{cases} 1 & x_n^2 + y_n^2 \leq 1 \\ 0 & otherwise \end{cases}$$

We expect that

$$\frac{\sum_{i=1}^{n} h_i}{n} \to \frac{\pi}{4}$$

Let $p = \frac{\pi}{4}$ and $p_n = \frac{h_n}{n}$. We then have

$$E[p_n - p] = E[p_n] - E[p]$$

$$= E[p_n] - p = E\Big[\frac{\sum_{i=1}^{n} h_i}{n}\Big] - p$$

$$= \frac{\sum_{i=1}^{n} E[h_i]}{n} - p = E[h_1] - p$$

$$= p - p = 0$$

Thus, p_n is an unbiased estimate. We then check the convergence rate of the sequence:

$$Var(p_n - p) = Var(p_n) - Var(p)$$

$$= Var(p_n) - 0 = \frac{\sum_{i=1}^{n} Var(h_i)}{n^2}$$

$$= \frac{Var(h_1)}{n}$$

$$= \frac{1}{n}\Big(E[h_1^2] - E[h_1]^2\Big)$$

$$= \frac{1}{n}\Big(\int_0^1 (1 - x^2)dx - (\frac{\pi}{4})^2\Big)$$

$$= \frac{1}{n}(\frac{2}{3} - (\frac{\pi}{4})^2)$$

Def: Let Y be a quantity, and let $\hat{Y}$ be an approximation of Y. Then, the absolute error is

$$e = \hat{Y} - Y$$

and the relative error is

$$\varepsilon = \frac{\hat{Y} - Y}{Y}$$

Note:

1 The sign is correct in the sense that e.g. if $\hat{Y} = 4$ and $Y = 3$ then $e = 1$ means that $\hat{Y}$ is an overestimate

2 $\hat{Y} = Y + e = Y + Y\varepsilon = Y(1 + \varepsilon)$, so $100 \cdot \varepsilon$ is the percentage error.